



IPT Guidance for Acquisition of Systems with Complex Programmable Hardware using DO-254



Contents

Introduction	3
Logic Devices	7
System Safety and Development Assurance	10
DO-254	16
Common Issues in DO-254 Development and Certification	23
References	30
Appendix A - Abbreviations	31
Appendix B - Glossary	32



SAFETY ENGINEERING

Terms of guidance:

This report is published by ERA Technology Limited, on behalf of the ASSC, to advance the role of standardisation in military systems. The use of this is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user.

Published 19 January 2009

Contacts

The ASSC is a unique MoD and industry forum bringing together a diverse network of commercial and military organisations. Through dedicated technical investigations, hand-on workshops and comprehensive web-resources, the ASSC provides its members, comprising largely of the UK MoD and its suppliers across land, sea and air domains, with the information required to best exploit applicable standards and technologies:

- Providing information and technical advice on current and proposed national and international standards.
- Influencing the standardisation processes.
- Encouraging, where appropriate, the harmonisation of military and civil standards.
- Promoting knowledge transfer.
- Providing expert advice on emergent technologies.

For more information and to register visit the ASSC website: www.assconline.co.uk or contact assc@era.co.uk.

Navigator Key

The process described in this brochure is complemented by additional information that has been presented in the following manner:



Information – Additional information provided for interest, including additional references.



Risks – Examples of pitfalls that need to be considered.

Introduction

The need for the Guidance

For UK military systems, the safety assurance of Complex Electronic Hardware (CEH) was specifically addressed by Def Stan 00-54 introduced in 1999 [1]. However, this standard was withdrawn in December 2004 and stated to be superseded by the much wider system level Def Stan 00-56 issue 3 [2]. The MoD intended that the less prescriptive approach to system safety assurance introduced by Def Stan 00-56 issue 3 would facilitate the development and certification of novel systems, including those with CEH.

The effect of this approach to safety assurance has been the removal of detailed guidance for the specification and procurement of safe CEH for UK military systems. For a rapidly developing technology, guidance is required for IPTs (Integrated Project Teams) and suppliers and its omission may discourage the exploitation of CEH due to the perception of increased project risk.

To address this deficiency, this booklet aims to guide the procurement and acceptance of military avionic systems. It is based on the continuing technical advances that are being made in electronic system design and the capabilities of Programmable Logic Devices (PLDs). There has been a great increase in the capability and potentiality of the following complex Integrated Circuit (IC) categories, listed in order of current importance:

Field Programmable Gate Arrays

There continues to be an exponential growth in the use of FPGAs to support microprocessor-based systems by the provision of “glue logic” to connect electronic components on a CBA and by handling high speed Input/Output (I/O) interfaces.

There is a huge potential (yet to be fully realised) for the replacement of a microprocessor or obsolete Application Specific Integrated Circuit (ASIC) with an FPGA, or, indeed, by the replacement of an obsolete circuit board with a single FPGA.

Complex Programmable Logic Devices (CPLDs)

Like FPGAs, these devices continue to be used to support microprocessor-based systems by handling high-speed I/O interfaces. The parallel computations to support these interfaces can be provided by CPLDs to release the microprocessor for other tasks.

Application Specific Integrated Circuits (ASICs)

The design of an ASIC enables maximum efficiency in the implementation of the chip to attain the highest performance requirements. ASICs are used only if the high cost of development can be justified.

These devices now provide mature implementation alternatives to microprocessor-based systems but the potential for design errors is the same as, if not greater than, conventional software based systems.



There is no widely accepted general term used for the type of Integrated Circuit (IC) addressed in this booklet. The term Logic Device (LD) or Complex Logic Device (CLD) has been used here. The Logic Devices divide into the following two types:

- PLDs (Programmable Logic Devices) and
- ASICs (Application Specific Integrated Circuits).

In practice, the term Programmable Logic Device (PLD) is occasionally used with the intention of including ASIC devices. The scope of these terms should be clarified in associated documentation. It should be noted that standards provide clarity by explicitly including ASICs when required e.g. "PLDs and ASICs".

The main difference between a PLD and an ASIC is that a PLD is manufactured as a standard device which is then "programmed" or configured for a specific application. A PLD may be programmed by placing the chip in a special "device programmer" or, increasingly, programmed without removal from the circuit board which is called ISP (In System Programming).

Aims of this Guidance

The objective of this document is to provide guidance for those aspects of the acceptance process for military avionic systems impacted by the use of Complex Electronic Hardware (CEH), particularly Programmable Logic Devices (PLDs). The guidance is particularly aimed at the Integrated Project Team (IPT) Duty Holder and the IPT safety management support staff, although the Contractor and other stakeholders may find the information useful.

The guidance concentrates on the process by which a strategy to achieve the required Design Assurance (See Glossary Appendix B) for these devices can be developed and implemented.

This guidance will help the IPT to work with other stakeholders (including its experts) to ensure the following aspects of procurement are addressed:

1. The design assurance requirements for embedded devices are identified and adequately specified.
2. The IPT, Design Authority, Safety Committee and other stakeholders work together to ensure the required design assurance evidence is produced in a timely and efficient manner.
3. The design assurance evidence presented for these embedded devices is sufficient to support the Safety Case for the system being acquired.

The current practice varies from IPT to IPT due to lack of adequate guidance. Indeed, some IPTs and Safety Committees are unaware that their platforms have systems containing PLDs. Furthermore, in some cases, Contractors have been unaware of their responsibilities under Def Stan 00-56 regarding these devices.

The basis of this guidance is that the DO-254, Design Assurance Guidance For Airborne Electronic Hardware [3], civil guidance produced to aid the certification of systems and equipment for civil aviation can be adapted for military acceptance.


Scope of the Guidance

This guidance is particularly aimed at the assurance of the design of programmable logic implemented on a single Integrated Circuit (IC) or “chip” such as a Field Programmable Gate Array (FPGA).

Design assurance is required at different levels of hardware design from the system level down to the individual components, such as Integrated Circuits, on a Printed Circuit Board (PCB). However, this document does not directly address the design of these Circuit Board Assemblies (CBAs) and Line Replaceable Units (LRUs) in which PLDs are embedded. In general, the Safety Case for this level of hardware design will be based on the current procedures and practices established by the Contractor.

Outline

Section 2 provides background information on Logic Devices, Section 3 provides information on Development Assurance in relation to System Safety Assessment, Section 4 provides an overview of DO-254, and Section 5 provides details on common issues facing DO-254 developments. Glossary and abbreviation lists are provided in the Appendices.

 It is expected that PLDs developed using the DO-254 guidelines will become increasingly common in embedded systems used in military applications. This expectation is based on the following trends and advantages:

1. There continues to be an exponential growth in the use of PLDs in all industrial sectors including military aerospace.
2. PLDs have particular advantages over microprocessor-based systems in terms of processing power, I/O capacity, reduced footprint and a reduction in the number of ICs on the circuit board.
3. The DO-254 guidance has been given increased prominence in the civil aerospace sector by the FAA Advisory Circular [4] and is becoming a “de facto” standard.
4. There is increased use of equipment with a civil heritage within the MoD and a determination to exploit civil standards where applicable.

Logic Devices

Programmable Logic Devices (PLDs)

A “general purpose” PLD may or may not be reprogrammable depending on the implementing technology. The programming is achieved by using SRAM or Flash technology (reprogrammable) or fuse/antifuse technology (not reprogrammable) to make or unmake the connections necessary to provide the required functionality.

The PLDs are usually divided into Simple PLDs (SPLDs) and Complex PLDs (CPLDs) with the term High Capacity PLD (HCPLD) sometimes used to distinguish the top end devices, which can support hundreds of I/O pins:

- The SPLDs are based on the original Programmable Logic Arrays (PLAs) and Programmable Array Logic (PALs). They are still occasionally used to implement very simple logic on a circuit board.
- The Complex PLDs (CPLDs) have evolved from the basic arrays of SPLDs by connecting logic arrays together within the same Integrated Circuit. Circuit board development by connection of SPLDs is now becoming possible within a single CPLD.

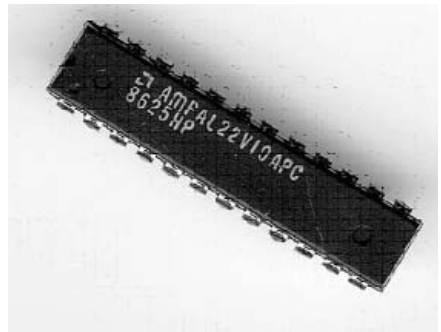


Figure 1: AMD 22V10 PAL Device – Example SPLD

Application Specific Integrated Circuits (ASICs)

An ASIC is manufactured by a long and expensive process in which the chip is fabricated in layers of silicon with minute embedded components and circuits. The connections on and between the layers provide the specific functionality for the application. For an ASIC, the logic is built into the device providing the opportunity for a more efficient and faster implementation. An ASIC cannot be reprogrammed.

Some engineers maintain that the logic is “programmed” into the device during manufacture as a basis for claiming an ASIC is a PLD. However, “programming” should be thought of as a separate process and by this criterion, an ASIC is not a Programmable Logic Device (PLD).

Three types of ASIC are generally distinguished:

Custom ASIC

A Custom (or Full-Custom) ASIC is designed by defining (virtually from scratch) all the photolithographic layers of the chip and their interconnections. The benefits of Custom Design usually include increased density (and therefore reduced area and recurring component cost), performance improvements and also the ability to integrate analogue components and other pre-designed (and potentially fully verified) components. The disadvantages can include increased manufacturing and design time, increased Non-Recurring Engineering (NRE) costs, more complexity in the Computer Aided Design (CAD) system and a much higher skill requirement on the part of the design team.

Cell based ASIC

Standard Cell design is the utilisation of functional blocks with known electrical characteristics, such as propagation delay, capacitance and inductance. These blocks can be represented in third party tools to achieve very high gate density and good electrical performance. Standard cell design fits between Gate Array and Full Custom design in terms of both its NRE and recurring component or unit cost.

Structured ASIC

Structured ASIC technology can be seen as bridging the gap between FPGA and "standard-cell" ASIC designs. They are based on the provision of pre-implemented standard layers leaving only a small number of chip layers that must be custom-produced. Structured ASIC designs have much smaller non-recurring expenditures (NRE) than "standard-cell" or "full-custom" chips, which require that a full mask set be produced for every design.

The predefined metal layers provide power, clock, and test structures that must be defined in other ASIC technologies and therefore can save time and expense for the designer.

Structured ASIC is being seriously addressed by commercial companies – albeit for the high volume and high performance market where time to market is important (e.g. for the telecommunications market rather than for military applications). Market analysts predict that "Structured ASIC" technology will continue to grow and overtake standard ASIC developments.

Field Programmable Gate Arrays (FPGAs)

FPGAs are technically PLDs when compared with ASICs. They have become so important they are sometimes distinguished from other PLDs. For example, FPGAs and PLDs should be read FPGAs and other PLDs.

There are 3 main families of FPGAs depending on how they are made: SRAM, Flash and anti-fuse. SRAM (Static RAM) based FPGAs have been dominant, benefiting from the advances of CMOS technology to keep ahead in terms of speed and density. However, SRAM is volatile unlike Flash and Anti-fuse. Flash based FPGAs are non-volatile and reprogrammable but lack the speed and density of SRAM FPGAs by one or two years. Anti-fuse FPGAs are not reprogrammable but are non-volatile and show greatest resistance to Single Event Upsets.

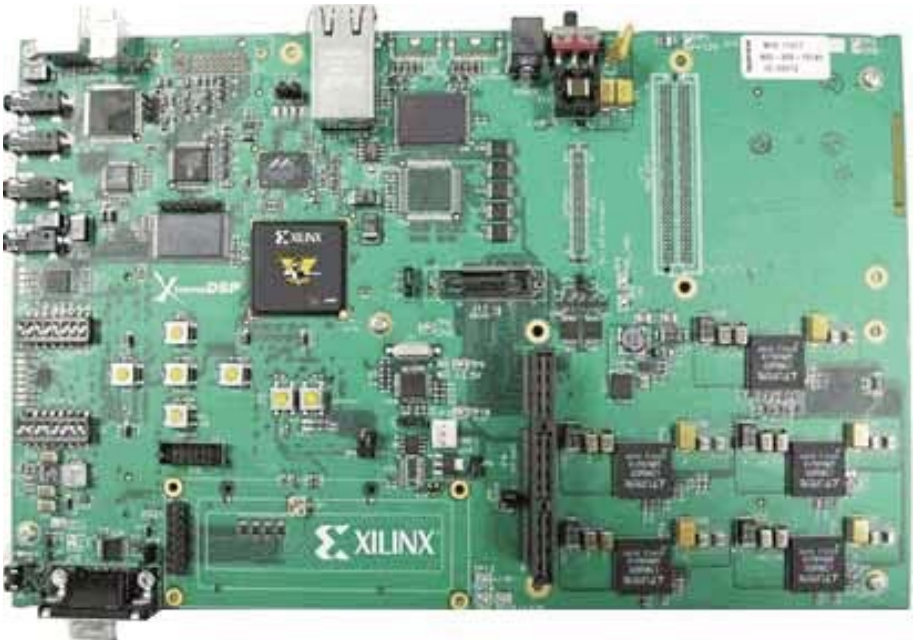


Figure 2: Xilinx Spartan
FPGA

System Safety and Development Assurance

In civil aerospace, the Safety Assessment of a system is considered to be a separate, parallel activity to its development, often undertaken by a different individual, team or even organisation. Nevertheless, the Safety Assessment is closely integrated with the design in that the results of Safety Assessment can affect the design, and the design decisions and activities are subject to Safety Assessment.

The Safety Assessment is based on hazard identification and analysis and associated risk assessment. Safety Assessment is conducted on the artefacts available at different stages of development during system design and during both hardware and software development. At each level of design, there may be different failure modes identified which must be analysed and the associated risks re-assessed. In civil aviation, the term "System Safety Assessment" includes consideration of the lower levels of hardware and software as necessary.

Level	Safety Assessment		Design Assurance	
	Defence	Civil Aviation	Defence	Civil Aviation
System	00-56	ARP 4761	00-56	ARP 4754
Software	00-55	ARP 4761	00-55	DO-178B
Hardware	00-54	ARP 4761 DO-254 Appendix B	00-54	DO-254

Table 1: Defence Standards and Civil Aviation guidelines

Safety Assessment – HAZID and analysis

For military systems, the Preliminary Hazard Identification (PHI) and PHA of Def Stan 00-56 result in "integrity" requirements on functions based on the risk reduction necessary to achieve "broadly acceptable" or "tolerable and ALARP" risks for the hazards associated with their failure.

In a similar way, in civil aviation, the Functional Hazard Assessment (FHA) and Preliminary System Safety Assessment (PSSA) assign a Development Assurance Level (DAL) to a function based on the severity of the worst associated aircraft failure condition associated with its anomalous behaviour.

The system design for both military and civil applications proceeds by allocating functions to subsystems and then to hardware and to software components within each subsystem. For civil systems, at each stage through the system, hardware and software design, "derived requirements" may be introduced which are passed to the Safety Assessment process for safety analysis, assessment and integration into the design at the appropriate level as described in ARP 4754 [7].

Design Assurance

The Design Assurance of a system is based on the demonstration that the functional requirements have been implemented to the appropriate level of rigour. The functional requirements arise from the following two sources:

1. System requirements concerned with the intended function of the system, and
2. Safety requirements to ensure that the system remains safe e.g. detection of unsafe states and undertaking appropriate corrective action.

The required rigour of the development of these functions is derived from the Safety Assessment and is expressed as a Design Assurance Level (DAL). For example, the Safety Assessment may determine that there are failure modes of the system functions, which may result in one or more hazards. The safety argument requires a certain assurance that these failures will not result from design errors. Similarly, for safety functions, the safety argument requires a certain assurance that the function will not fail to detect the unsafe state and will not fail to perform the correct action due to design error. Design Assurance activities are undertaken at different stages in a development, during system design and during both hardware and software development.

There is an explicit assumption in the application of DO-254 and DO-178B [5] that there is a System Safety Assessment (outside the scope of DO-254 or DO-178B) by which a DAL is assigned to the CEH device and a Software Level is assigned to the software. In civil aviation, guidelines and methods for conducting the System Safety Assessment Process are described in ARP 4761, Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment [6]. This area is also addressed in Def Stan 00-56. The restricted scope of DO-254 and DO-178B to design assurance should be recognised.



The relationship between Hardware Safety Assessment and Hardware Design Assurance is so important that Appendix B of DO-254 is dedicated to describing an approach to Hardware Safety Assessment, namely Functional Failure Path Analysis (FFPA), that can be used to develop a Hardware Design Assurance strategy required for DAL A and B hardware. The FFPA is essentially the application of system safety assessment as described in ARP 4761 at various levels of hardware design.

DALs

DALs are defined in the civil aviation ARP 4754 at system level. They depend on the worst credible “Failure Condition” at aircraft level that can result from the use of a system, including its failure modes.

A “Failure Condition” is used in civil aviation to refer an aircraft level functional “hazard”, that is, a hazardous state which results from a functional failure of a system or device. A “Failure Condition” is classified by an estimation of the combination of the likely reductions in safety margins and the ability of the flight crew to cope with adverse operating conditions and possible adverse effects on occupants. Indeed, the “worst credible effects” is subject to interpretation.

The “Failure Condition” is given a classification of “Catastrophic”, “Hazardous”, “Major”, “Minor” or “No effect”. In operation, the classification of the “Failure Condition” may escalate as safety margins are reduced by the unavailability of backup systems. The “Failure Condition Classification” then results in the DAL requirement of A, B, C, D or E on the system or device associated with functional failure.

The definition of “Failure Condition” classification as the “worst credible consequence with which a failure could be associated” often depends on engineering judgement about failure containment measures.

An IPT Duty Holder is required to agree a “Safety Integrity” scheme such as DALs with the Contractor to specify the required “reliability of each safety function” in addition to the specification of the safety function itself.

Failure Condition Classification	System/Device DAL
Catastrophic	A
Hazardous	B
Major	C
Minor	D
No Effect	E

Table 2: DAL Assignment

The use of any safety integrity scheme (including DALs) for the required reliability of a safety function should focus on the following issues:

1. What argument and evidence will be used to establish the safety integrity requirements (possibly expressed as a DAL) for the system, hardware and/or software development?
2. What argument and evidence will be used to demonstrate the safety integrity requirements (possibly expressed as a DAL) have been achieved?
3. For military systems adapted from civil aviation, are there fundamentally different safety requirements that may require a different approach to the design assurance requirements? For example, much different safety integrity requirements may derive from the consideration of transition to war (or training for war) scenarios. In these cases, what extra evidence is required to achieve the design assurance required?

The DAL results from the System Safety Assessment, which determines the criticality of the system. It is from this that the criticalities of its subsystems and their hardware and software are derived.

The setting of the design assurance requirements for a system in civil aviation is usually the result of a consensus of engineering judgement after previous assessment and the feedback of experience from the operation of generations of similar systems. This vast experience has not always been captured and presented in a formal argument.

Relationship between Integrity Schemes

There may be a requirement to consider the relationship between DALs and other safety integrity schemes. For example, the safety functions of a PDS for an upgrade may have used the Safety Integrity Level (SIL) scheme of Def Stan 00-56 issue 2. Conversely, a legacy platform may still be contracted against Def Stan 00-56 issue 2 and may want to upgrade with a civil certified system.

There are no general rules for the correspondence of safety integrity schemes for systematic failures. Each case must be treated separately and the safety assessments and evidence behind the safety integrity schemes must be examined carefully and extra safety work undertaken if necessary.

With the introduction of Def Stan 00-56 issue 3/4, the emphasis has been on the direct setting of objectives for the development and subsequently demonstrating that these objectives have been met rather than working indirectly through Safety Integrity Levels.

It is important to understand that DALs at system level are a relatively recent introduction for civil aviation. The software guidelines DO-178/A/B introduced the idea of Software Level to describe different levels of design assurance. However, the guidance has always referred to the Software Level being determined following a system safety assessment process outside the scope of the guidelines.

It was only in 1996 that the Aerospace Recommended Practices, ARP 4754 for the development of “complex or highly integrated aircraft systems” introduced the concept of Development Assurance Levels at system level. In addition ARP 4761 presented recommended techniques for undertaking system safety assessments by which these levels could be set or justified.

Furthermore, it should be noted that these documents were only gradually accepted by the civil aviation community and have only recently been recognised by reference in FARs and CSs. They do not appear in Advisory Circulars stating that their application is an acceptable means of compliance with the requirements of FAR/CS 25-1309 [8] for System Analysis and Design.

There is a formal process by which a DAL is derived but the evidence for compliance to this process is often unavailable for older systems. However, for more recent systems, there is much more likely to be a formal argument for a particular DAL.

The process of deriving and assigning of DALs has been a stumbling block for many users of DO-254. An FAA memorandum outlining draft Policy on the Guidance for Determination of System, Hardware and Software Development Assurance Levels has been published [9]. This recognises that the relationship between ARP 4754, DO-178B and DO-254 is not seamless with respect to the determination of development assurance levels. In the absence of consistent guidance the policy aims to provide a standardised approach to the use and application of these guidelines and any associated industry practices.

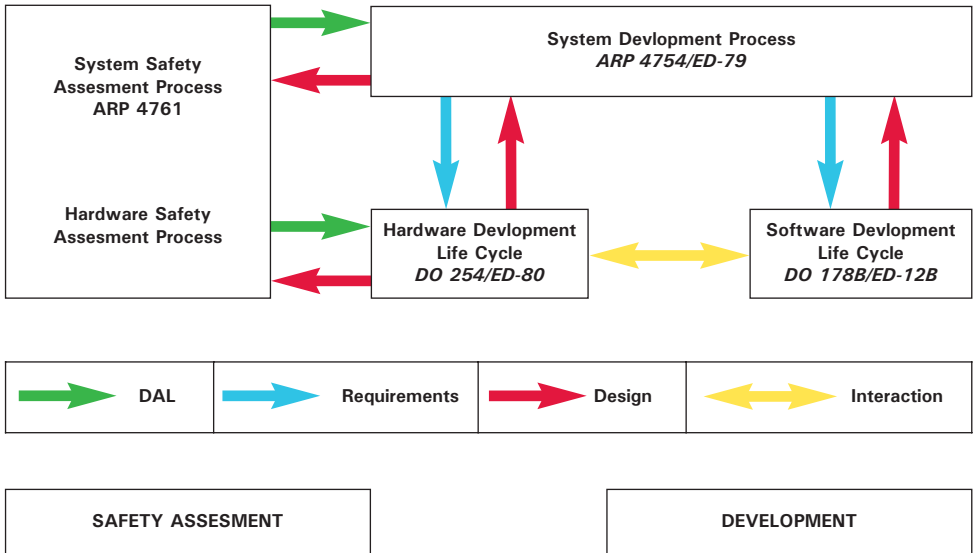



Figure 3: Civil Aviation Development Processes and Guidelines

DO-254

An IPT Duty Holder may be faced with the following questions:

1. How can the safety integrity requirements of the airborne electronic hardware of a system be specified using DO-254 in a way that is compatible with Def Stan 00-56 issue 3/4, in particular Part 2 Section 2? For example, what are the limitations and risks associated with the specification of demonstration of compliance to DO-254?
2. What does a claim of compliance of a hardware development to DO-254 by a supplier mean in terms of Def Stan 00-56 issue 3/4? Will extra hardware design assurance be required to meet the safety integrity requirements of Def Stan 00-56 issue 3/4?

This section provides an overview of DO-254 to help the IPT formulate informed answers.



DO-254 aims to assist organisations by providing design assurance guidance for the development (and certification) of airborne electronic hardware. The design assurance is an important contribution to the demonstration that the system will safely perform its intended function in its specified environment as required by the Federal Aviation Regulations (FARs) and equivalent EASA Certification Specifications (CSs) [8]. The important sections are FAR/CS 25.1301 (intended function) and FAR/CS 25.1309 (safety) for large transport aircraft and equivalent parts for other aircraft types. The design assurance evidence must be complemented by a safety assessment.

Scope of DO-254

The full application of DO-254 means gaining design assurance for all levels of the CEH development by a rigorous application of a number of processes in a lifecycle and the collation and assessment of associated evidence.

The guidance within the DO-254 document, itself, is stated to be applicable, but not limited, to the following hardware items:

1. "Custom micro-coded components, such as Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs) and Programmable Logic Devices (PLDs)".
2. "Integrated technology components, such as hybrids and multi-chip modules".
3. "Commercial-Off-The-Shelf (COTS) components".
4. "Circuit Board Assemblies (CBAs)".
5. "Line Replaceable Units (LRUs)".

For civil certification, DO-254 is only formally used for item 1 above, that is PLDs in the widest sense to include FPGAs and ASICs. The Advisory Circular [4] has had the effect of both endorsing and limiting the use of DO-254 to these "custom micro-coded" devices.

It is important to distinguish the following scopes for the application DO-254 guidance:

1. DO-254 can be applied to the development of CEH at all levels of integration from Integrated Circuits to Circuit Board Assemblies and LRUs. This is the full scope as stated in DO-254 itself.
2. DO-254 can be restricted to the design of PLDs (including FPGAs and ASICs).

It should be recognised that a practical interpretation of the DO-254 guidelines is still being developed between certifiers and developers. There are many issues that are still being resolved for the certification of civil aviation systems. These include the following issues:

1. The particular requirements for qualification of Tools for CEH development.
2. The assurance requirements for the integration of previously developed hardware into a hardware design.

These are similar issues to those addressed by software, but more difficult as both the hardware and software are programmed in a PLD. It is the resolution of these issues which will facilitate the successful exploitation of PLD technology in both military and civil avionics. It is advised that the arguments used for civil certification are examined for applicability and possible adaptation for military systems.

Principles of DO-254

DO-254 is based on the following principles for the hardware development:

1. A hardware development lifecycle must be established
2. The guidance is presented with reference to basic “standard” life cycle processes such as “planning”, “requirements capture”, “design”, and “implementation” which must be undertaken in any development.
3. For each development, the developer must provide a “mapping” or explanation of each standard process in terms of the specific life cycle processes presented for the development.
4. The guidance is given for each of the standard processes in terms of the following:
 - Objectives for the process
 - Activities – guidance (recommendations) for achieving the objectives.
5. Hardware “Design Assurance Levels (DALs)” are used to differentiate the rigour of design assurance evidence required for a development.

- 6. DO-254 provides guidance on the Hardware Safety Assessment
- 7. The DO-254 guidance emphasises the iterative nature of Hardware Safety Assessment and Hardware Design whereby “derived requirements” from the Hardware Design are fed back for safety assessment.
- 8. Life Cycle Data must be generated during the hardware development.

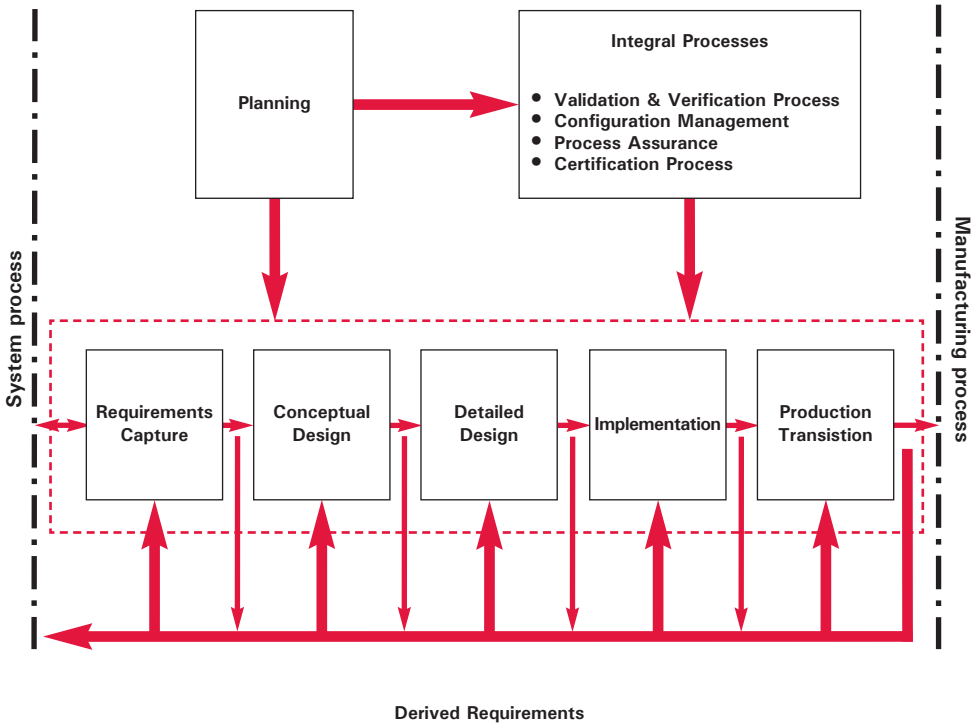


Figure 4: DO-254 Hardware Development Processes

Hardware Safety Assessment and Strategy

DO-254 provides important guidance about the Hardware Safety Assessment process (for setting safety integrity requirements) and its relationship to the Hardware Design Assurance process (the subject of DO-254) for meeting those requirements. The Safety Assessment of the hardware arises naturally as part of the System Safety Assessment (SSA) process.

Top down methods such as FTA, which analyse a functional failure at system level in terms of the failures of subsystems down to the failures of hardware. Similarly, there are bottom up methods such as FMEA; a FMEA starts from each known or possible failure mode of a hardware item and analyses the wider effect of the failure.

The related Failure Mode, Effects and Criticality Analysis (FMECA) emphasises that the criticality of the consequences of the failure is also examined.

The Hardware Safety Assessment of a hardware item is an assessment of the contribution of its failure (or anomalous behaviour) to a system hazard and the possible development of an aircraft level “failure condition”.

The Hardware Safety Analysis is conducted as part of the System Safety Analysis (SSA) using “conventional safety techniques” in conjunction with the Functional Failure Path Analysis (FFPA) technique described in Appendix B of DO-254. The “path” terminology is derived from circuit analysis and refers to the sequence of failures by which a single failure can escalate to become a failure at a higher level, possibly even an aircraft failure condition. This analysis results in DALs being assigned to the hardware circuits or components whose failure or anomalous behaviour can contribute to one or more Functional Failure Paths.

The Hardware Safety Assurance Strategy is based on the results of the Hardware Safety Assessment, and depends on the identification and analysis of the paths by which the hardware can fail: the Functional Failure Paths (FFPs) in DO-254. The Hardware Design Assurance Strategy for each circuit or component then depends on the DAL assigned and whether the hardware item (circuit or component) is considered to be “complex” or “simple” as described in the glossary. This is described in Table 3.

Hardware DAL/Complexity	Hardware Design Assurance Strategy
High Integrity (Level A/B) & Complex	<p>Full Strategy required using one or more of:</p> <ul style="list-style-type: none"> • advanced analysis, • product service history, • architectural mitigation
High Integrity (Level A/B) & Simple OR Medium Integrity (Level C)	<p>“Applicable fail-safe aspects” addressed: Measures should be put in place to eliminate or mitigate the consequences of the identified failures and these must be documented together with the associated safety analyses such as common mode analyses. The guidelines also require an analysis to ensure that the measures introduced to ensure safety, have not compromised the other performance requirements of the hardware item.</p>
Low or no integrity requirement (Level D/E)	<p>No guidance given.</p> <p>This does not mean that Hardware Safety Assurance does not have to be addressed (even at DAL E the analysis that there are no safety related failures will have to be maintained and revisited when there are changes).</p> <p>There will be Hardware Design Assurance requirements to support the development at DAL D or E. The AC [4] explicitly states that the developer may choose to continue to use his own procedures for Hardware Design Assurance at Level D. However, the Hardware Design Assurance approach must be agreed with the certification authority.</p>

Table 3: Hardware Design Assurance Strategy

Def Stan 00-56 and Design Assurance

Def Stan 00-56 issue 3/4 [2] is an evidence based standard and the following points are noted:

1. The full application of DO-254 should provide the necessary design assurance evidence for PLDs. On their own, the alternative sources of evidence such as history of previous use and the quality of the fabrication process will not be strong enough at medium and high levels of integrity.
2. A DO-254 compliant process should provide the necessary evidence for the design assurance of the other hardware items above. However, there may be an alternative argument and evidence that the required design assurance or integrity is achieved. The argument and evidence put forward for each system must be assessed separately.

A developer's claim that DO-254 certification of electronic hardware has been achieved usually means the following:

1. The required design assurance has been achieved for the PLDs and ASICs as described in DO-254, and
2. The design assurance for the higher levels of integration of the hardware has been achieved using the developer's established methods for electronic circuit design; however, it should be noted that some companies have introduced procedures to support the full scope of DO-254.

Common Issues in DO-254 Development and Certification

This section presents a number of issues that have been encountered during the acquisition of systems developed using the DO-254 guidance for CEH in civil aviation applications. It should be noted that most of these issues have also arisen during the acquisition of systems with software developed to the DO-178B guidelines.

Several of these issues could have been addressed by early agreement with the certification authority. In military terms, the certification authority is the IPT Duty Holder, supported by the Safety Committee, the Independent Safety Auditor (ISA), where applicable, and the Release to Service Authority (RTSA).

For the procurement of programmable hardware for military aviation, it is important that there is a mechanism for raising issues and agreeing solutions early in the programme, and that this mechanism is used. The Plan for Hardware Aspects of Certification (PHAC) or military equivalent should be considered for this use in military procurement.

Inadequate Level of Detail in Requirements

There is evidence accumulated from many projects that more operational failures are caused by the misunderstanding of requirements than by the incorrect implementation of requirements. For complex hardware, the problem may be traced to the development organisation that is responsible for the decomposition of system requirements and their translation into hardware (and software) requirements. Nevertheless, IPTs can encourage reviews of CEH requirement specifications (and software requirement specifications) and focus attention by requesting attendance at review meetings for the IPT and/or its technical representative(s).



An inadequate level of detail in requirements seems common to all complex acquisitions. The initial experience with the acquisition of complex electronic hardware has shown the same weakness in poor requirements specification.

The following steps have been found to be useful:

1. Establishing a robust “Requirements Review” process

The argument is often made that there has been an extensive internal review of requirements within the MoD before award of contract and the designer has demonstrated that these requirements are understood in his proposal, so further “interference” by the IPT is not necessary or helpful, especially at the level of CEH. However, the current view of the requirements engineering discipline is that expensive errors can be avoided by an appropriate review process at all levels of design.

2. Establishing contacts with the actual Development Team (for bespoke development)

The individuals in the development organisation who responded to the Invitation to Tender (ITT) may not be the engineers tasked with the actual development. These developers may not have a clear mechanism, or the necessary contacts, to obtain clarification of requirements. The IPT should be prepared to re-establish understanding, especially with developers in a large organisation.

Inadequate Formal Planning and following of Plans

Although planning is a continuous process throughout an acquisition, a plan should be substantially complete (albeit subject to revision and expansion) before the associated work is started. Furthermore, plans should be approved by the Quality Assurance¹ (QA) representative, the certification authority and the IPT or its representative(s) before work is started. In many acquisitions, this approval may be delegated to the Design Authority although the IPT can exert influence through project reporting and Safety Panel meetings.

In civil aviation, the late presentation of the PHAC and/or PSAC is a major cause of late projects and project cost overruns as technical work is required to be repeated or extra unplanned work is required when the plans are eventually reviewed. Furthermore, certification authorities find that the presented plans are often not compliant with DO-254 for CEH or with DO-178B for software developments (even though compliance is claimed).

¹Quality Assurance is referred to as Process Assurance in DO-254.

The following steps have been found to be useful:

1. An IPT should encourage developers to produce and use checklists to ensure that the plans and subsequent developments meet the guidance. The developers should produce their own checklists to address their own unique environment and to provide a mechanism to feed back lessons learned from other projects. Nevertheless, an IPT can encourage the developers to adopt best practice in several ways, such as by encouraging third party audits.
2. As regards following the plans, an IPT should require oversight of the developer's internal review process, self-auditing and QA results. In addition to the regular reporting by the developer, the MoD may require attendance (or attendance by its technical representative) at major reviews and for higher integrity projects, may require auditing of the development by a third party.

Lack of Independence in Quality Assurance and Verification

The requirements for "independence" are often misunderstood and consequently "independence" is not always achieved when required. The different requirements for "independence" of QA from development and for "independence" of verification from design, are a source of misunderstanding.

Quality assurance

In civil aviation standards, the quality assurance "function" is required to be "independent" of the development at all DO-254 design assurance levels. In this context, "independent" refers to the relationship between the QA representative and the development team. As regards reviews, the QA representative assures that the work has been reviewed according to the QA plan. The plan may or may not require an "independent" review by technically competent people who have not been involved in the work (see next section).

The IPT should encourage the Design Authority to audit the quality management systems of its suppliers for specific projects. The oversight role is taken by the FAA or CAA (delegated by EASA) in civil aviation; the CAA can ask to attend a Design Authority audit of a supplier as an observer, or, if there are specific concerns, the CAA can lead the audit of the supplier.

For a small company, independence may be established if there is more than one project and a member on one project provides the QA role for another project. In each case, the arrangements to establish and maintain independence should be documented. Larger companies usually have a special QA department that provides a QA representative for each project. The QA department should have a different reporting route up to board level from the development project.



Small companies often have a problem with respect to establishing independence of the quality assurance function as all the staff may be committed to a single project in one role or another. The quality assurance is often claimed to be provided by a technical manager who reviews and signs off all the work. In these cases, the required level of independence is often not achieved as the manager has been involved in some aspects of the development.

A large programme can be put under risk of failing certification if the Design Authority does not ensure that the suppliers implement adequate quality management systems with sufficient independence from the development.

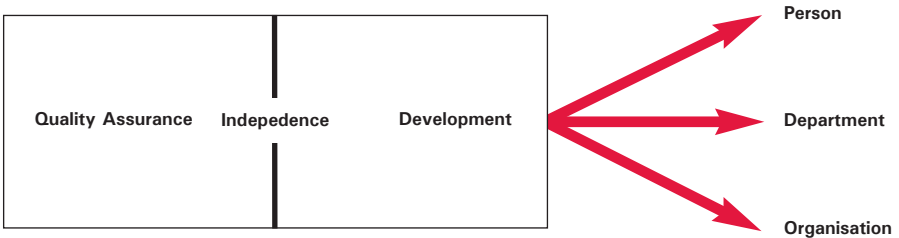


Figure 5: QA Independent of Development

Verification

For DO-254 (and DO-178B), verification refers to the actual technical details of establishing by review, analysis and testing that the results of a development stage are correct with respect to requirements. Furthermore, DO-254, Appendix A, describes independence as a “means to address potential common mode errors... and should be performed with an individual, process or a tool that is independent of the designer”.

The following points should be noted from Appendix A of DO-254:

1. DO-254 requires all verification of Level A and B functions to be independent but “Level C and lower functions do not require independent verification”
2. “Organisational structure separation is not required to achieve independence”.

With regard to verification by testing, “independence from design” refers to the planning, specification and the execution of the tests as well as the review of the test results. For example, for Levels A and B, the tests are usually planned, produced, executed and the results reviewed by a test team independently of the CEH development. In these cases, the developers and testers work independently of each other but from the same requirements specification.

Finally, DO-178B (and DO-254 by close association) has the concept of “verification of verification”, whereby the approach of the verification is technically checked and it is confirmed that the test results really do show that the requirements are satisfied. This is a final technical check, independent from the QA check, that the procedures for review have been followed.

Inadequate and non-automated Traceability

“Traceability” refers to the establishment of connections from a “requirement” at a certain development level (for example, from a system, hardware or software “requirement”) possibly through intermediate levels to the following:

1. The test(s) that verify its implementation, and
2. The design decompositions and representations down to the hardware and/or software by which the requirement is implemented.

The establishment of traceability is a prerequisite for a full justification that the requirement is verified by the test/s and that the combination of hardware and software does implement the requirement.

Depending on the design assurance level required, it may only be required to establish partial traceability. For example, at DAL D, traceability of each hardware requirement to its test/s is required, but traceability through the design to its implementation is not required.

A traceability tool can perform checks to support the claim that the traceability is provided for each requirement. In general, the lack of an automated traceability tool can affect the programme. It is generally more difficult to maintain and demonstrate full traceability when the changes are performed “by hand”. In these cases, the effort to demonstrate complete traceability at key milestones can be considerable.

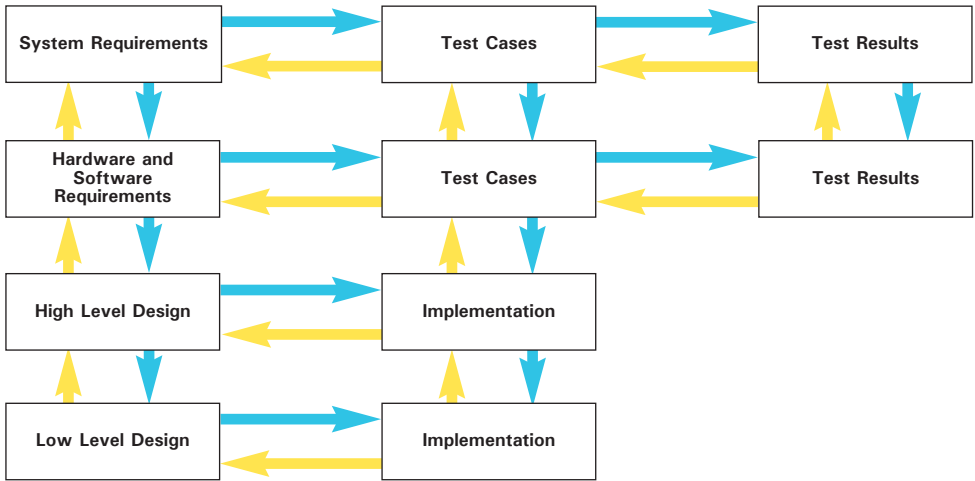


Figure 6: Forwards and Backwards Traceability

It is a common problem that traceability cannot be demonstrated because there are problems with the configuration management of the various documents involved such as requirements, design, tests, test environment and implementation.

Even with the correct versions of documents, it is often the case that the traceability of a requirement is inadequate. For example, some of the tests to ensure that the requirement is completely verified have not been identified, or some of hardware or software modules essential for its implementation have been omitted from the traceability data.

For complex developments, a developer sometimes claims traceability has been established for each requirement on the basis of a list of tests and a list of hardware/software modules, but without a documented rationale for the list of tests, this claim is unfounded.

Lack of Automated Testing

It is often necessary to repeat extensive testing even after a small change. The amount of retesting may be reduced as a result of an impact analysis but for modern CEH, a small change can result in large changes to the implementation.

For example, the algorithms applied by a “place-and-route” tool may result in a large reorganisation as optimisation is sought. Many projects have found that several, or even many more, testing iterations have been required than originally planned.

The following experiences are given of real projects (thought to be typical) where the testing of the CEH was not automated.

1. A relatively small, development team under deadline and cost pressures determined that there was not enough effort and time to automate the tests. The tests were performed “by hand” on the naïve expectation that they would not have to be repeated (at least before the first delivery and its associated payment). In practice the tests had to be repeated several times before acceptance and the extra effort and cost in automation would have been more than recouped. The tests are now being automated in order to support the hardware in the long term.
2. For a high DAL development, an inexperienced team had been unsure of the qualification requirements for a tool to automate its testing, and had avoided this certification risk by performing the tests “by hand”. The effort turned out to be very expensive as the testing was extensive and had to be repeated. The issue should have been raised early with the certification authority and the qualification plan for the tool could have been documented in the PHAC and agreed with the certification authority. The development team found that the certification authority were concerned that an automated tool had not been used.

References

1. Requirements for Safety Related Electronic Hardware in Defence Systems
Def Stan 00-54 interim, Mar 1999 withdrawn December 2004
2. Safety Management Requirements for Defence Systems
Def Stan 00-56 issue 2, December 1996
Def Stan 00-56 issue 3 (interim), Dec 2004
Def Stan 00-56 issue 4, June 2007
3. Design Assurance Guidance For Airborne Electronic Hardware
DO-254, RTCA (16/04/2000)
ED-80, EUROCAE
4. Subject: DO-254 - Design Assurance Guidance For Airborne Electronic Hardware
Advisory Circular 20-152
Federal Aviation Authority (30/06/05)
5. Software Consideration in the Certification of Airborne Systems and Equipment
DO-178B, RTCA, December 1992
ED-12B, EUROCAE
6. Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, ARP 4761
Society of Automotive Engineers (SAE) - (12/1996)
7. Certification Considerations for Highly-Integrated or Complex Aircraft Systems, vARP 4754, ED-79
Society of Automotive Engineers (SAE) - (10/4/1996)
8. FAR/CS 25
Certification Specifications, including airworthiness codes and acceptable means of compliance, for large aeroplanes
9. Policy Statement on Guidance for Determination of System, Hardware, and Software Development Assurance Levels on Transport Category Airplanes ANM-03-117-09 (Policy Memorandum)
US Department of Transport, 15th January 2004.

Abbreviations

AC	Advisory Circular (from FAA)
ALARP	As Low As Reasonable Practicable
ARP	Aerospace Recommended Practice (produced by SAE for FAA)
ARP 4754	Civil Aviation Guidelines for developing complex systems
ARP 4761	Civil Aviation Guidelines for undertaking Safety Assessment
ASIC	Application Specific Integrated Circuit.
CAA	Civil Aviation Authority (see EASA)
CCA	Common Cause Analysis
CEE	Complex Electronic Element – PLD or ASIC (including microprocessor)
CEH	Complex Electronic Hardware
CLB	Configurable Logic Block – part of FPGA architecture
CLD	Complex Logic Device
COTS	Commercial-Off-The-Shelf
CPE	Complex Programmable Element (software or complex electronic hardware)
CPLD	Complex PLD usually made up of several SPLDs
CS	Certification Specification (equivalent to FAR) of EASA
DAL	Development/Design Assurance Level
EASA	European Aviation Safety Agency
EUROCAE	European Civil Aviation Equipment organisation
FAA	Federal Aviation Authority
FAR	Federal Aviation Regulation (equivalent to EASA's Certification Specification)
FFP	Functional Failure Paths
FFPA	Functional Failure Path Analysis
FHA	Functional Hazard Assessment
FMEA	Failure Mode and Effects Analysis
FPGA	Field Programmable Gate Array
FTA	Fault Tree Analysis
HAS	Hardware Accomplishment Summary
HDL	Hardware Description Language
I/O	Input/Output
IC	Integrated Circuit
IPT	Integrated Project Team
LRU	Line Replaceable Unit
NRE	Non Recurring Engineering /Expenditure
PCB	Printed Circuit Board (cf. CBA)
PHA	Preliminary Hazard Analysis
PHAC	Plan for the Hardware Aspects of Certification
PHI	Preliminary Hazard Identification
PLA	Programmable Logic Array (SPLD with 2 programmable planes of logic gates)
PLD	Programmable Logic Device (general term including PLA, PALs, FPGAs)
QA	Quality Assurance
RTCA	Originally Radio Technical Commission for Aeronautics, now just RTCA
RTSA	Release to Service Authority
SAE	Society of Automotive Engineers (produce ARPs)
SIL	Safety Integrity Level
SPLD	Simple PLD
SSA	System Safety Assessment

Term	Definition	Reference
Application Specific Integrated Circuit (ASIC)	An Integrated Circuit developed for a specific application. Its production involves a long development time and large setup (NRE) costs as the transistors and connections on each layer of silicon and between layers on the chip are defined and laid out by teams of engineers. The benefit is optimised performance and ultimately cost if enough units can be sold to offset the large set up costs. It is becoming increasingly rare for ASICs to be developed purely for military purposes as required performance can be achieved by PLDs.	
Complex	A complex system is one in which it is not feasible to test all possible inputs and cannot be fully understood without the use of analytical tools c.f. Simple.	RTCA DO-254
Complex Electronic Element	An element of a system that is implemented in software or custom hardware (i.e. by PLD)	Def Stan 00-56 Part 2 Issue ¼
Configurable Logic Block	Basic configurable block of Inputs and Outputs in a CPLD or FPGA. Also macrocell.	
Configuration Identification	The process of defining and designating a Configuration Item.	RTCA DO-254
Configuration Item	One or more components, tools or data items treated as a unit for configuration management purposes.	RTCA DO-254
Configuration Management	<p>(1) The process of Configuration Identification, and the control of issues and changes of Configuration Identities.</p> <p>(2) A discipline applying technical and administrative direction and surveillance to identify and record the functional and physical characteristics of a configuration item, control changes to those characteristics, and record and report change control processing and implementation status.</p>	RTCA DO-254
Core	<p>A Core is a self contained function of an IC usually an FPGA or ASIC.</p> <p>A soft core (or IP Core) is a function described by its logic function rather than a physical implementation. It is usually presented in terms of a Hardware Description Language (HDL) to be integrated with other HDL for other functions.</p> <p>A hard core is a physical implementation of a function sometimes referred to as embedded cores.</p>	

Term	Definition	Reference
Design Process	<p>The process of creating a hardware (or software or system) item from a set of requirements. For hardware design, DO-254 explicitly names the following processes : requirements capture, conceptual design, detailed design, implementation and production transition.</p> <p>N.B. The meaning of "Design" depends on the context. It can mean</p> <ol style="list-style-type: none"> 1) just the conceptual design and detailed design processes 2) the set of processes listed above excluding other supporting processes such as verification and configuration management 3) all the processes required to create the item (the general term "Development" is often instead of Design). 	RTCA DO-254
Design Assurance	<p>All of those planned and systematic actions used to substantiate, at an adequate level of confidence, that design errors have been identified and corrected such that the system/hardware/software satisfies the application certification basis.</p> <p>N.B. Development Assurance is same definition with "design errors" replaced by "development errors"</p>	RTCA DO-254
Design Assurance Level` (DAL)	<p>The DAL is an assurance level A, B, C, D or E assigned to hardware functions by the system safety assessment based on the worst credible failure condition e.g. DAL A is assigned to hardware functions whose failure or anomalous behaviour, as shown by the hardware safety assessment, would cause a failure of system function resulting in a catastrophic "failure condition" for the aircraft.</p>	RTCA DO-254 Table 2.1
Development Assurance	<p>All of those planned and systematic actions used to substantiate, at an adequate level of confidence, that development errors have been identified and corrected such that the system/hardware/software satisfies the application certification basis. N.B. Design Assurance is same definition with "development errors" replaced by "design errors".</p>	SAE ARP 4754
Hardware Design Analysis	Term not used in DO-254 but part of development	
Hardware Design Assurance	See Design Assurance – the required level of assurance is expressed as a (Hardware) Design Assurance Level derived from the System Safety Assessment (to which the Hardware Safety Assessment contribute)	RTCADO-254

Term	Definition	Reference
Hardware Safety Assessment	See Safety Assessment. The Hardware Safety Assessment is part of the System Safety Assessment.	
Hardware Safety Assurance	See Safety Assurance. The Hardware Safety Assurance requirements are translated into Hardware Design Assurance requirements.	
Integral Processes	<p>Those processes that ensure the correctness, control, and confidence of the hardware life cycle processes and their outputs. The integral processes are the verification process, the configuration management process, the quality assurance process, and the certification liaison process.</p> <p>The integral processes are performed concurrently with the hardware development process.</p>	From the definition in RTCA DO-178B which refers specifically to the software life cycle
Programmable Array Logic (PAL)	Simple PLD produced from modifying PLA by fixing one of the programmable planes. First produced in 1978.	
Programmable Logic Array (PLA)	Simple PLD produced by the Signetics company which evolved to become Xilinx which has been the leader in CPLDs and FPGAs for the last twenty years. First produced in 1977.	
Safety Analysis	<p>1) An investigation of the component parts of a safety related aspect of a system, product, process and their relations in contributing to the whole. e.g. Fault Tree Analysis</p> <p>2) Used as a general term to cover all investigations except review and testing.</p>	Based on dictionary definition of "analysis"
Safety Assurance	The result of planned and systematic actions necessary to provide adequate confidence that a product or process satisfies safety requirements.	Adapted from definition of "assurance" in RTCA DO-254
Safety Assessment	An evaluation based on engineering judgement that a product or process satisfies safety requirements.	Adapted from definition of "assessment" in RTCA DO-254
Simple	A Simple System is one which is not complex i.e. it is feasible to test all possible inputs and it can be fully understood without the use of analytical tools	RTCA DO-254

Term	Definition	Reference
Validation	The process of determining that the requirements are the correct requirements and that they are complete. In DO-254, the (hardware) validation process does not address the initial validation of the system requirements - these are validated at system level. The hardware validation process addresses the validation of "derived requirements" arising during the hardware design.	RTCA DO-254
Verification	The evaluation of an implementation of requirements to determine that they have been met.	RTCA DO-254